**INTRODUCTION**

Some people can imagine that the digital processing of image means only processes of adornment of images and the introduction of some of the decorations and drawings on them and their intransigence to appear in another appearance is different from the original ,but the digital processing of images beyond that ,but in fact can include the processes of pressure and images optimization whether digital or medical. the most important application in the field of digital image processing using waveguide, is its application in compression and encoding of digital images(image compression and coding)at present, photo compression technology is known as enabling technology to handle large volume of data and send it, (spatial resolution) especially in the field of sensor imaging, medical application, database creation, aerodynamic application and remote sensing. Every day, the amount of stored information that is processed and transmitted digitally increases. The compression method is therefore the best solution to reduce the size of the data required to represent the digital image in the form of to dimensional matrix consisting of a group of image cells consisting of a set of data the digital image compression technology has pushed forward the image processing field, and multimedia development as well as progress in the internet, making the internet accessible to people with access and access. Li any site.

subsequently decompressing the input image.

 Fig. 1 Basic steps in an image compression system.

The compression that is achieved can be quantified by the

compression ratio given by the following formula:

 C

R

=n

1

/n

2

 (1)

where n

1

 and n

2

 denote the number of information

carrying units (bits) in the original image and the

compressed image respectively. A compression ratio like

10 (or 10:1) indicates that the original image has 10

information carrying units (e.g. bits) for every 1 unit in the

compressed data set.

Several quality measurement variables like, PSNR (peak

signal-to-noise ratio), MSE (mean square error) etc. can be

measured to find out how well an image is reproduced

with respect to the reference image. These variables are

signal fidelity metrics and do not measure how viewers

perceive impairments. Numerical values of these variables

for any image tell us about the quality of that image [6-8].

The measure of peak signal-to-noise ratio (PSNR) is

defined as the following formula:

PSNR = 10log

10

(255

2

 / MSE) dB (2) (2)

and, mean square error (MSE) is given by,

The Wavelet Transform

Wavelets are signals which are local in time and scale and

generally have an irregular shape. A wavelet is a

waveform of effectively limited duration that has an

average value of zero. The term ‘wavelet’ comes from the

fact that they integrate to zero; they wave up and down

across the axis. Many wavelets also display a property

ideal for compact signal representation: orthogonality.

This property ensures that data is not over represented. A

signal can be decomposed into many shifted and scaled

representations of the original mother wavelet. A wavelet

transform can be used to decompose a signal into

component wavelets. Once this is done the coefficients of

the wavelets can be decimated to remove some of the

details. Wavelets have the great advantage of being able to

separate the fine details in a signal. Very small wavelets

can be used to isolate very fine details in a signal, while

very large wavelets can identify coarse details. In addition,

there are many different wavelets to choose from. Various

types of wavelets are: Morlet, Daubechies, etc

. [9, 10].

One particular wavelet may generate a more sparse

representation of a signal than another, so different kinds

of wavelets must be examined to see which is most suited

to image compression.

A wavelet function Ψ(t) has two main properties,

0

;0)( dtt

That is, the function is oscillatory or has wavy appearance.

0

2

;)( dtt

That is, the most of the energy in Ψ(t) is confined to

*Chapter one*

1.1. image type:

**1. TIFF (also known as TIF), file types ending in .tiff**

TIFF stands for Tagged Image File Format.[14] TIFF images create very large file sizes. TIFF images are uncompressed and thus contain a lot of detailed image data (which is why the files are so big) TIFFs are also extremely flexible in terms of color (they can be grayscale, or CMYK for print, or RGB for web) and content (layers, image tags).

TIFF is the most common file type used in photo software (such as Photoshop), as well as page layout software (such as Quark and InDesign), again because a TIFF contains a lot of image data.

**2. JPEG (also known as JPG), file types ending in .jpg**

JPEG stands for Joint Photographic Experts Group, which created this standard for this type of image formatting.[9] JPEG files are images that have been compressed to store a lot of information in a small-size file. Most digital cameras store photos in JPEG format, because then you can take more photos on one camera card than you can with other formats.

A JPEG is compressed in a way that loses some of the image detail during the compression in order to make the file small (and thus called “lossy” compression).

JPEG files are usually used for photographs on the web, because they create a small file that is easily loaded on a web page and also looks good.

JPEG files are bad for line drawings or logos or graphics, as the compression makes them look “bitmappy” (jagged lines instead of straight ones).

**3. GIF, file types ending in .gif**

GIF stands for Graphic Interchange Format. This format compresses images but, as different from JPEG, the compression is lossless (no detail is lost in the compression, but the file can’t be made as small as a JPEG).[13]

GIFs also have an extremely limited color range suitable for the web but not for printing. This format is never used for photography, because of the limited number of colors. GIFs can also be used for animations.

**4. PNG, file types ending in .png**

PNG stands for Portable Network Graphics. It was created as an open format to replace GIF, because the patent for GIF was owned by one company and nobody else wanted to pay licensing fees. It also allows for a full range of color and better compression.[12]

It’s used almost exclusively for web images, never for print images. For photographs, PNG is not as good as JPEG, because it creates a larger file. But for images with some text, or line art, it’s better, because the images look less “bitmappy.”

When you take a screenshot on your Mac, the resulting image is a PNG–probably because most screenshots are a mix of images and text.

**5. Raw image files**

Raw image files contain data from a digital camera (usually). The files are called raw because they haven’t been processed and therefore can’t be edited or printed yet. There are a lot of different raw formats–each camera company often has its own proprietary format.[11]

1.2.Image compression algorithm:

**EZW Algorithm**

An EZW encoder is specially designed to use with wavelet transforms [23]. The is based on progressive coding to compress an image into bit stream with increasing accuracy then more bits are added to a bit stream image will be more detailed.

The wavelet transform uses filter banks for the decomposition of preprocessed image [24]. The EZW encoder encodes the decomposed image by recognizing the priority of decomposed image pixel. The coding algorithm includes the first step to

determine the initial threshold, if we choose bit plane coding then initial threshold *T*0 will be *T*0 = 2[*log* (*max* ((*U* (*x, y*)))]. *MAX* (*U*

(*X*, *Y*)) is the maximum coefficient with this threshold [25]. We analysis peak signal to noise ratio and mean square error by using this coding algorithm.

The **encoding process** is done using two passes. The dominant pass generates any one of four possible combinations are significant positive (SP), significant negative (SN), isolated zero (IZ) and zero tree root (ZR). Subordinate pass where the

coefficients are encoded as 0 or 1 depending on the current threshold [23].



 Figure 3. Block diagram for EZW

The **decoding process** is done using two passes. The decoding unit reconstructs the values by identifying the symbols as positive, negative, zero tree and isolated zero tree. Inverse transformation is the process of retrieving back the image data from the obtained image values. The image data transformed and decomposed under encoding side is rearranged from higher level decomposition to lower level with the highest decomposed level been arranged at the top Fig3 shows the reconstruction of the obtained decomposed component [26].

**2.1.2 WDR Algorithm**

 WDR is another direct approach to find the distances of these significant coefficients bits are compressed[26]. It consists of five steps:

**1. Initialization:** During this step an assignment of a scan order should first be made. For an image with P pixels, a scan order is a one-to-one and onto mapping between the wavelet coefficient and a linear ordering. The scan order is a zigzag through sub bands from higher to lower levels. For coefficients in sub bands they are two methods of scanning are there one is row based another is column based. In row-based scanning is used in the horizontal sub bands and column based scanning is used in the vertical sub bands. The scan order is zigzag scanning is used for the diagonal and low-pass sub bands. As the scanning order is made, an initial threshold is chosen so that all the transform

values satisfy |*Xm*|<*T*0 and at least one transform

value satisfies |*Xm*|> = *T*0 / 2.

**2. Update threshold:** Let *Tk*= *Tk*−1/ 2.

**3. Significance Pass:** In this part, transform values are deemed significant if they are greater than or equal to the threshold value. Then their index values are encoded using the DWR. The difference reduction method essentially consists of a binary encoding of the number of steps to go from the index of the last significant value to the index of the current significant value.

The output from the significance pass includes the signs of significant values along with sequences of bits, generated by difference reduction, which describes the precise locations of significant values.

**4. Refinement Pass:** The refinement pass is to generate the refined bits via the standard bit-plane quantization procedure.

Each refined value is a better approximation of an exact transform value.

**5**. **Repeat steps** (26) through (25) until the bit budget is reached.



Figure 2. Block diagram for WDR

3.1.2.Huffman algorithm:

The Huffman Coding is a lossless data compression algorithm, developed by David Huffman in the early of 50s while he was a PhD student at MIT. The algorithm is based on a binary-tree frequency-sorting method that allow encode any message sequence into shorter encoded messages and a method to reassemble into the original message without losing any data.

### **The Basics**

The algorithm is based on the frequency of occurrence of the data item(byte). The most frequent data items will represented and encoded with a lower number of bits.

The main idea of the algorithm is create a binary tree, called Huffman tree, based on the bytes frequency on the data, where the leafs are the bytes symbols, and the path from the root to a leaf determines the new representation of that leaf byte.

### **Building the Tree**

Each node of the tree are represented with a byte symbol and the frequency of that byte on the data. The creation of the Huffman tree have the following steps:

* **Scan the data and calculate the frequency of occurrence of each byte;**
* **Insert those nodes into a reverse priority queue based on the frequencies(a lowest frequency is given highest priority);**
* **Start a loop until the queue is empty;**
* **Remove two nodes from the queue and combine them into a internal node with the frequency equal to the sum of the two nodes frequencies;**
* **Insert the two nodes removed from the queue as children of the created internal node;**
* **Insert the created internal node into the queue;**
* **The last node remaining on the queue is the root of the tree**.

Using the text **ABCBAACD**asexample and applying those steps, we have the following tree:



Generated with: <http://huffman.ooz.ie/>

So the new representation of the bytes on the text are:

* A: 0
* B: 10
* C: 111
* D=110

**Conclusion**:

The original representation has 8 bytes(64 bits) and the new representation have only 9 bits, that is **86%** **smaller** than the original. So the Huffman Coding turns to be a simple and efficient way to encode data into a short representations without loosing any piece of information.

The idea with this post is introduce to this compression algorithm, and on the next posts show the implementation and creation of a compression library using Ruby

***:* 4.1.2.DWT algorithm**

In everyday life, when one hears “wavelet”, they understand a “small water wave”. They do not wander off to things like image compression for instance. However, in mathematics, wavelets refer to short wavelike functions that can be scaled and translated. They are called wavelets due to their characteristic of integrating to 0, “waving” up and down the x − axis. In fact, wavelet transforms can take any signal and express it based on those scaled and translated wavelets. The result of this procedure is a representation of the given signal at different scales [1]. Wavelet transforms are definitely very important computational tools. A transform is a familiar concept to mathematicians. It is a widely used standard mathematical tool that helps with solving problems in multiple areas. The fundamental idea of transforms is changing a mathematical quantity (it could be a number, a vector, a function, etc.) to another form where it may be unrecognizable, but would present useful features. This transformed quantity is, thus, used to solve the problem on hand, or to perform some helpful calculation. The result can then be transformed back to the original form [2, p.104]. Initially, wavelets were solely in mathematics. Now, the extent of their usage has reached 4 areas such as seismology, image processing, quantum mechanics, signal processing, nonstationary signals in particular, and data compression. Among their applications in real-life, we can give two examples: *Figure 1:*’cameram an.tif’ famous test image The 8 × 8 matrix I representing that corner is:9



156 159 158 155 158 156 159 158 160 154 157 158 157 159 158 158 156 159 158 155 158 156 159 158 160 154 157 158 157 159 158 158 156 153 155 159 159 155 156 155 155 155 155 157 156 159 152 158 156 153 157 156 153 155 154 155 159 159 156 158 156 159 157 161 

 Compressing this matrix or the whole image follows the same algorithm that we applied to the array r. We just need to repeat the procedure for the rest of the matrix’s rows. Then do the same for its columns. The resulting matrix looks like columns. The resulting matrix looks like:



*Figure2*: : The original cameraman image and its lossless version

156.86 −0.2031 −0.1562 −0.0625 0.75 −0.1875 −0.25 −0.5 −0.25 −0.5 0.6406 −0.1718 0.2812 −0.3125 0 0.6875 0.25 0.75 0 0 0 0 0 0 0 0 −0.2812 −0.2812 −0.4375 0.75 0 −0.625 0.75 0 −0.125 0 0.375 −0.375 −2.25 1 1 0.25 −0.125 0 0.375 −0.375 −2.25 1 1 0.25 0.06250 0.0625 −0.375 −0.25 0.75 −0.5 1.75 1.75 −1.625 0.375 −1 0.25 0.75 0.75 0.25 0.75 As we can see, the resulting matrix has multiple 0 entries, and most of the other entries are actually close to 0. This result can be, attributed to the usage of differencing, as well as 10 to the fact that usually, an images adjacent pixels do not differ by much. Our process can be implemented using loops in order to go over all the rows and columns of the image, which is a bit tedious. However, we can simplify this and make it faster using linear algebra. It may seem daunting to use linear algebra but the process is pretty straightforward. Matrix multiplication is the key her

I∗ Because of the large number and size of fingerprint images, the FBI developed a fingerprint compression specification called Wavelet Scalar Quantization which is based on wavelet compression [3]. ∗ In order to determine accurately the redshifts of galaxies, it is necessary to identify the key lines of their spectra. This identification problem needs to be tackled in an automated and reliable manner because it requires large sky surveys which produce a huge volume of data. Consequently, a wavelet-based method, called the Darth Fader algorithm, has been established for estimating redshifts of galaxy spectra [4]. Wavelet transforms are employed profusely in image processing and compression. Actually, they enable computers to store images in many scales of resolution. They decompose a given image into a number of details and approximations. Since many of the compression processes are quite similar to each other, investigating any one algorithm is enough to get a lot of insight into the field of image compression as a whole. Therefore, through this capstone project, focus will be on the Haar wavelet transform, its usage in image compression, as well as the performance of its different variants different variants. mage Compression Using Discrete Wavelet Transforms.pdf

Image Compression Using the Haar Wavelet Transform Actually, the array r that we have compressed earlier is the first row of the upper left 8 × 8 corner of a well-known grayscale image. It is the standard cameraman test image. Courtesy of the Massachusetts Institute of Technology.

1.3.definition of compression:

### **data compression is a reduction in the number of**[**bits**](https://whatis.techtarget.com/definition/bit-binary-digit)**needed to represent data. compressing data can save storage capacity, speed up file transfer, and decrease costs for storage hardware and network**[**bandwidth**](https://searchnetworking.techtarget.com/definition/bandwidth).

### **How compression works**

is performed by a program that uses a formula or [algorithm](https://whatis.techtarget.com/definition/algorithm) to determine how to shrink the size of the data. For instance, an algorithm may represent a string of bits -- or 0s and 1s -- with a smaller string of 0s and 1s by using a dictionary for the conversion between them, or the formula may insert a reference or pointer to a string of 0s and 1s that the program has already seen.

Text compression can be as simple as removing all unneeded [characters](https://whatis.techtarget.com/definition/character), inserting a single repeat character to indicate a string of repeated characters and substituting a smaller bit string for a frequently occurring bit string. Data compression can reduce a text file to 50% or a significantly higher percentage of its original size.

### For data transmission, compression can be performed on the data content or on the entire transmission unit, including [header](https://whatis.techtarget.com/definition/header) data. When information is sent or received via the internet, larger files, either singly or with others as part of an [archive](https://searchstorage.techtarget.com/definition/archive) file, may be transmitted in a ZIP, [GZIP](https://searchdatacenter.techtarget.com/definition/gzip-GNU-zip) or other compressed format.

### Why is data compression important?

Data compression can dramatically decrease the amount of storage a file takes up. For example, in a 2:1 compression ratio, a 20 megabyte ([MB](https://searchstorage.techtarget.com/definition/megabyte)) file takes up 10 MB of space. As a result of compression, administrators spend less money and less time on storage.

Compression optimizes backup storage performance and has recently shown up in [primary storage data reduction](https://searchstorage.techtarget.com/definition/data-reduction-in-primary-storage-DRIPS). Compression will be an important method of data reduction as data continues to grow exponentially.

Virtually any type of file can be compressed, but it's important to follow best practices when choosing which ones to compress. For example, some files may already come compressed, so compressing those files would not have a significant impact.

### **1.4type compression methods: lossless and lossy compression**

Compressing data can be a [lossless or lossy](https://whatis.techtarget.com/definition/lossless-and-lossy-compression) process. Lossless compression enables the [restoration](https://searchdatabackup.techtarget.com/definition/restore) of a file to its original state, without the loss of a single bit of data, when the file is uncompressed. Lossless compression is the typical approach with executables, as well as text and spreadsheet files, where the loss of words or numbers would change the information.

Lossy compression permanently eliminates bits of data that are redundant, unimportant or imperceptible. Lossy compression is useful with graphics, audio, video and images, where the removal of some data bits has little or no discernible effect on the representation of the content.

Professor David Brailsford, with the School of
Computer Science at the University of Nottingham,
discusses compression of text and pictures.Graphics [image compression](https://whatis.techtarget.com/definition/image-compression) can be lossy or lossless. Graphic image file formats are typically designed to compress information since the files tend to be large. JPEG is an image file format that supports lossy image compression. Formats such as GIF and PNG use lossless compression.

**Chapter**

**tow**

 **: 1.5.main feature of EZW**

      The main features of EZW include compact multiresolution representation of images by discrete wavelet transformation, zerotree coding of the significant wavelet coefficients providing compact binary maps, successive approximation quantization of the wavelet coefficients, adaptive multilevel arithmetic coding, and capability of meeting an exact target bit rate with corresponding rate distortion function (RDF). The details of this algorithm can be found in ([[6]](https://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/MITRA/REFERE.HTM#[6])). This algorithm may not yield optimal distortion but it does provide a practical and general high compression algorithm for a variety of image classes.

      The core of the EZW compression is the exploitation of self-similarity across different scales of an image wavelet transform. In other words EZW approximates higher frequency coefficients of a wavelet transformed image. Because the wavelet transform coefficients contain information about both spatial and frequency content of an image, discarding a high-frequency coefficient leads to some image degradation in a particular location of the restored image rather then across the whole image. As with other wavelet-based techniques, it also does not introduce blocking artifacts, inherent in windowed-frequency plane based compression methods such as JPEG ([[21]](https://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/MITRA/REFERE.HTM%22%20%5Cl%20%22%5B21%5D)). Another property of wavelet transforms is that for a wide class of images the wavelet coefficients tend to decrease in magnitude as we go to finer scales of the image. At the same time the number of the finer level coefficients grows as 22j where j is the number of decomposition levels. The larger coefficients require more bits to be represented than smaller ones. As it can be seen more frequent coefficients (finer scale coefficients) require less bits for their representation, which is the underlying principle of entropy coding. In the original EZW algorithm introduced by Shapiro this property was not used efficiently. If this property is properly exploited, as it has been suggested by Amir Said and William Pearlman ([[1]](https://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/MITRA/REFERE.HTM%22%20%5Cl%20%22%5B1%5D)), not only does the EZW algorithm efficiently approximate the details of an image, but it also allocates more bits for less frequent components, implicitly implementing entropy coding, while executing the main EZW algorithm. This could lead to further reduction in the execution time if the follow-up arithmetic coding used in the original EZW algorithm which does not provide a significant increase in the compression ratio, is eliminated.

      This new modified EZW algorithm can be very valuable in image transmission via the Internet, because the compression/decompression execution time decreases as the compressio

n ratio gets higher allowing for rough-scale image previews before the image is downloaded. However, AFLC-VQ is also capable of fast decoding following a progressive transmission scheme with a 400 MHz Pentium or equivalent as a client computer of data that is uncompressed.

Because of this, the size of a raw file is extremely large. Usually they are converted to TIFF before editing and color-correcting

1.6.concpet of EZW:

The basic idea of ​​the configuration of the zeros tree that starts from the root and to the end of the branches is that many unimportant transactions in the high frequency sub-packets can be neglected The planted zero tree encodes the structure of the tree so that the resulting transactions can be represented in a gradual manner [20] The representation of the algorithm and the formation of the zero tree require three basic phases. Note Figure 1. [16]

****

**Figure 1.The basic stages of representing the EZW algorithm:**

1. Wavelet Transform: The image is analyzed into A number of sub-packets and multiple levels, which results in the distribution of transactions in packets so that hierarchical transactions are correlated in a hierarchical way The hierarchical representation of transactions is an important characteristic of cloning applications in which important parameters are determined in high frequency packets [22] [15]. 2. Scalar Quantization: At this stage, the data representing the waveguide coefficients of the image are reduced using numerical quantification resulting in loss of some information to the image of the clamping process. [16] Wavelet Transform of original image Scalar Quantization of wavelet coefficients.

3. Entropy Encoding : This process is performed on all packages and by using similarity between the adjacent transactions and all the levels in each direction. These transactions are encoded in descending order, using The Zig-Zag Scanning Order. identifies the importance of wave transactions in this stage, encapsulating each zero tree containing the non-important coefficients expressed in zero (0), and the important coefficients expressed in 1 . [17] leading to reduced data, which For something that helps maintain the quality

The image at loopback. [19]

****

**Figure 2.The Scan Scan Zag-Zig method on all transaction locations**

1.7.EZW algorithm:

The embedded zero tree wavelet algorithm (EZW) is a simple, yet remarkable effective, image compression algorithm, having the property that the bits in the bit stream are generated in order of importance, yielding a fully embedded code. Using an embedded coding algorithm, an encoder can terminate the encoding at any point thereby allowing a target rate or target distortion metric to be met exactly. Also, given a bit stream, the decoder can cease decoding at any point in the bit stream and still produce exactly the same image that would have been encoded at the bit rate corresponding to the truncated stream. In addition to producing a fully embedded bit stream, EZW consistently produces compression results that are competitive with virtually all known compression algorithms.

A wavelet coefficient *x* is said to be insignificant with respect to a given threshold *T* if |*x*|<*T*. The zerotree is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a threshold, then all wavelet coefficients of the same orientation in the same spatial location at the finer scale are likely to be insignificant with respect to the same threshold. More specifically, in a hierarchical sub band system, with the exception of the highest frequency sub bands, ever coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation. The coefficient at the coarse scale is called the *parent,*and all coefficients corresponding to the same spatial location at the next finer scale of similar orientation are called *children.*Similar, we can define the concepts *descendants*and *ancestors .*The data structure of the zero tree can be visualized in Figure ![[*]](). Given a threshold *T* to determine whether or not a coefficient is significant, a coefficient *x* is said to be an element of a *zero tree*for the threshold *T* if itself and all of its descendents are insignificant with respect to the threshold *T*. Therefore, given a threshold, any wavelet coefficient could be represented in one of the four data types: zero tree root (ZRT), isolated zero (IZ) (it is insignificant but its descendant is not), positive significant (POS) and negative significant (NEG).



***Figure: Coefficients are coded in a zero tree structure and scanned in a left-to-right order.***

**1.7.EZW encoding*:***

When searching through wavelet literature for image compression schemes it is almost impossible not to note Shapiro’s Embedded Zerotree Wavelet encoder or EZWencoder for short [Sha93]. An EZW encoder is an encoder specially designed to use with wavelet transforms, which explains why it has the word wavelet in its name. The EZW encoder was originally designed to operate on images (2D-signals) but it can also be used on other dimensional signals. The EZW encoder is based on progressive encoding to compress an image into a bit stream with increasing accuracy. This means that when more bits are added to the stream, the decoded image will contain more detail, a property similar to JPEG encoded images. It is also similar to the representation of a number like π. Every digit we add increases the accuracy of the number, but we can stop at any accuracy we like. Progressive encoding is also known as embedded encoding, which explains the E in EZW. This leaves us with the Z. This letter is a bit more complicated to explain, but I will give it a try in the next paragraph. Coding an image using the EZW scheme, together with some optimizations results in a remarkably effective image compressor with the property that the compressed data stream can have any bit rate desired. Any bit rate is only possible if there is information loss somewhere so that the compressor is lossy. However, lossless compression is also possible with an EZW encoder, but of course with less spectacular results.

The EZW encoder is based on two important observations: 1. Natural images in general have a low pass spectrum. When an image is wavelet transformed the energy in the sub bands decreases as the scale decreases (low scale means high resolution), so the wavelet coefficients will, on Embedded Zero tree Wavelet Encoding – © C. Valens, 1999 – c.valens@mindless.com 5 average, be smaller in the higher sub bands than in the lower sub bands. This shows that progressive encoding is a very natural choice for compressing wavelet transformed images, since the higher sub bands only add detail; 2. Large wavelet coefficients are more important than smaller wavelet coefficients. These two observations are exploited by the EZW encoding scheme by coding the coefficients in decreasing order, in several passes. For every pass a threshold is chosen against which all the coefficients are measured. If a wavelet coefficient is larger than the threshold it is encoded and removed from the image, if it is smaller it is left for the next pass. When all the wavelet coefficients have been visited the threshold is lowered and the image is scanned again to add more detail to the already encoded image. This process is repeated until all the wavelet coefficients have been encoded completely or another criterion has been satisfied (maximum bit rate for instance). The trick is now to use the dependency between the wavelet coefficients across different scales to efficiently encode large parts of the image which are below the current threshold. It is here where the zero tree enters. So let me now add some detail to the foregoing. (As most explanations, this explanation is a progressive one.) A wavelet transform transforms a signal from the time domain to the joint time-scale domain. This means that the wavelet coefficients are two-dimensional. If we want to compress the transformed signal we have to code not only the coefficient values, but also their position in time. When the signal is an image then the position in time is better expressed as the position in space. After wavelet transforming an image we can represent it using trees because of the sub sampling that is performed in the transform. A coefficient in a low sub band can be thought of as having four descendants in the next higher sub band (see figure 1). The four descendants each also have four descendants in the next higher sub band and we see a quad-tree emerge: every root has four leafs.

******

 **Figure 1 The relations between wavelet coefficients in different sub bands as quad-trees.**

\*. Example In [Sha93] an incomplete example is given. Here we will give the complete output stream of the algorithm described above. The example data is shown in figure 3.

****

**Figure 3 The example data from [Sha93] together with two scan orders. We used the Morton scan order.**

\* Notes • The implementation presented here was on a high level and leaves room for many optimizations;

• EZW encoding does not really compress anything, it only reorders wavelet coefficients in such a way that they can be compressed very efficiently.

 An EZW encoder should therefore always be followed by a symbol encoder, for instance an arithmetic encoder (as in [Sha93]);

•Loss EZW compression/decompression is very similar to wavelet shrinkage. I will add an explanation of this later (I hope).

**Chapter**

***Three***

***1.9.practical application of EZW algorithm:***

 ***1.PSNR***

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image.

The Mean Square Error (MSE) and the Peak Signal to Ratio (PSNR) Noise are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

To compute the PSNR, the block first calculates the mean-squared error using the following equation:

*MSE*=Ξ*M*,*N*[*I*1(*m*,*n*)−*I*2(*m*,*n*)]2*M*∗*N*

In the previous equation, M and N are the number of rows and columns in the input images, respectively. Then the block computes the PSNR using the following equation:

*PSNR*=10log10(*R*2*MSE*)

In the previous equation, R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

### Recommendation for Computing PSNR for Color Images

Different approaches exist for computing the PSNR of a color image. Because the human eye is most sensitive to luma information, compute the PSNR for color images by converting the image to a color space that separates the intensity (luma) channel, such as YCbCr. The Y (luma), in YCbCr represents a weighted average of R, G, and B. G is given the most weight, again because the human eye perceives it most easily. With this consideration, compute the PSNR only on the luma channel.

****

**2.compresssion ratio:**

After doing some processing on the image, say DCT and Quantization, how can I calculate the compression ratio between the compressed image and the original one??

Let A be the storage size needed to represent the original image. For example use whos() to fetch it.

Let B be total storage size of all the arrays together that are needed to hold the compression information that would be needed to recover the image. For dct type arrangements be sure to include the storage needed to indicate which coefficients have been zeroed.

Now A/B is your compression ratio. If you get a result less than 1.0 then that means that your compression representation takes more space than the original, which does happen for some data and some compression algorithmsSo if I get a result more than 1.0, that means my compression is good? and what should I do to get it in percentage? for example if I want to say that I've compressed this image (A) by 30%

**3.mean squared error:**

Mean-squared error

## Syntax

err = immse(X,Y)

## Description

[example](https://www.mathworks.com/help/images/ref/immse.html#buicn9m-6)

[err](https://www.mathworks.com/help/images/ref/immse.html#buicn9l-1-err) = immse([X](https://www.mathworks.com/help/images/ref/immse.html#buicn9l-1-X),[Y](https://www.mathworks.com/help/images/ref/immse.html#buicn9l-1-Y)) calculates the mean-squared error (MSE) between the arrays X and Y. X and Y can be arrays of any dimension, but must be of the same size and class.

## Examples

collapse all

### Calculate Mean-Squared Error in Noisy Image

Try This Example

Read image and display it.

ref = imread('pout.tif');

imshow(ref)

****

Create another image by adding noise to a copy of the reference image.

A = imnoise(ref,'salt & pepper', 0.02);

imshow(A)

****

Calculate mean-squared error between the two images.

err = immse(A, ref);

fprintf('\n The mean-squared error is %0.4f\n', err);

 The mean-squared error is 353.7631

**4.DWT**

Wavelets are signals which are local in time and scale and generally have an irregular shape. A wavelet is a waveform of effectively limited duration that has an average value of zero. The term ‘wavelet’ comes from the fact that they integrate to zero; they wave up and down across the axis. Many wavelets also display a property ideal for compact signal representation: orthogonality. This property ensures that data is not over represented. A signal can be decomposed into many shifted and scaled representations of the original mother wavelet. A wavelet transform can be used to decompose a signal into component wavelets. Once this is done the coefficients of the wavelets can be decimated to remove some of the details. Wavelets have the great advantage of being able to separate the fine details in a signal. Very small wavelets can be used to isolate very fine details in a signal, while very large wavelets can identify coarse details. In addition, there are many different wavelets to choose from. Various types of wavelets are: Morlet, Daubechies, etc. [9, 10]. One particular wavelet may generate a more sparse representation of a signal than another, so different kinds of wavelets must be examined to see which is most suited to image compression. A wavel. ∫ ∞− ∞4. Proposed Compression Method using DWT This section illustrates the proposed compression technique with pruning proposal based on discrete wavelet transform (DWT). The proposed technique first decomposes an image into coefficients called sub-bands and then the resulting coefficients are compared with a IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012 ISSN (Online): 1694-0814 www.IJCSI.org 328 Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved. threshold. Coefficients below the threshold are set to zero. Finally, the coefficients above the threshold value are encoded with a loss less compression technique. The compression features of a given wavelet basis are primarily linked to the relative scarceness of the wavelet domain representation for the signal. The notion behind compression is based on the concept that the regular signal component can be accurately approximated using the following elements: a small number of approximation coefficients (at a suitably chosen level) and some of the detail coefficients. Fig. 2 The structure of the wavelet transform based compression. The steps of the proposed compression algorithm based on DWT are described below: I. Decompose Choose a wavelet; choose a level N. Compute the wavelet. Decompose the signals at level N. II. Threshold detail coefficients For each level from 1 to N, a threshold is selected and hard thresholding is applied to the detail coefficients. III. Reconstruct Compute wavelet reconstruction using the original approximation coefficients of level N and the modified detail coefficients of levels from 1 to et function Ψ(t) has two main properties, ∫ ∞− =Ψ 0 dtt ;0)( That is, the function is oscillatory or has wavy appearance N.

****

Fig.1 A real image and corresponding compressed images with GIF, JPEG and proposed DWT methodsThe experimental results with the proposed compression method have been arranged in the Table 1 for different threshold values. From this table, we find that a threshold value of δ = 30 is a good choice on the basis of trade-off for different compression ratios. Table 2 shows the comparison between JPEG, GIF and the proposed compression method. Experimental results demonstrate that the proposed compres Table 1. Compression Results with Proposed Method for different threshold values sion technique gives better performance compared to other compression techniques..

Table 1. Compression Results with Proposed Method for different threshold values ****

***8. Conclusions***

A new image compression scheme based on discrete wavelet transform is proposed in this research which provides sufficient high compression ratios with no appreciable degradation of image quality. The effectiveness and robustness of this approach has been justified using a set of real images. The images are taken with a digital camera (OLYMPUS LI-40C). To demonstrate the performance of the proposed method, a comparison between the proposed technique and other common compression techniques has been revealed. From the experimental results it is evident that, the proposed compression technique gives better performance compared to other traditional techniques. Wavelets are better suited to time-limited data and wavelet based compression technique maintains better image quality by reducing errors. The future direction of this research is to

input image  That results from compressing

**10.REFREANC**

[[1]](https://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/MITRA/INTRO.HTM#REF9) Antonini, M., Barlaud, M., Mathieu, P. and Daubechies, I., "Image coding using wavelet transform," IEEE Transactions on Image Processing, Vol. 1, No. 2, pp. 205-220,1992.

 [2] Newton, S.C., Pemmaraju, S. and Mitra, S., " Adaptive fuzzy leader clustering of complex datasets in pattern recognition," IEEE Transactions on Neural Networks, vol. 3, no. 5, pp. 784-800, 1992.

 [3] Mitra, S. and Pemmaraju, S., "Efficient color image compression using integrated fuzzy neural networks for vector quantization," Invited paper presented at the SMC Symosium, Orlando, Florida, October 12-17, 1997.

[4] Meadows, S. and Mitra, S., "Low Bit Rate Color Image Coding with Adaptive Encoding of Wavelet Coefficients," Data Compression Conference, DCC,97, Snowbird, Utah, 25-27 March 1997.

 [5] Mitra, S. and Pemmaraju, S., "Adaptive Vector Quantization using an ART-based Neuro-fuzzy Clustering Algorithm," Invited Paper presented at the International Conference on Neural Networks, June 3-6, Washington, D. C., June 3-6, 1996.

[6] Mitra, S. and Yang, S.-Y., " Adaptive Vector Quantization for High Fidelity Color Image Coding at Very Low Bit Rates," accepted for publication in Journal of Electronic Imaging, SPIE, January 1999.

[7] Special Issue on Vector Quantization, IEEE Transactions on Image Processing, Vol.5, No. 2, February, 1996.

 [8] Shapiro, J.M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," IEEE Transactions on Signal Processing, vol. 41, no. 12, December 1993.

[9] Pennebaker, W.B. and Mitchell, J.L., "JPEG Still Image Data Compression Standard," Van Nostrand Reinhold, New York, 1993.

[[10]](https://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/MITRA/INTRO.HTM#REF1) Said, A. and Pearlman, W.A., " A new fast and efficient image codec based on set partitioning in hierarchial trees," IEEE Transactions on Circuits and Systems for Video Technology, 6, 243-250, 1996.

[11] Foley, James D., van Dam, Andries, Feiner, Steven K., and Hughes John F.,

*Computer Graphics Principles and Practice,* Addison-Wesley, Reading, MA,

1996.

[12] Boutell, Thomas et al., "PNG Specification," Version 1.0, PNG Development

Group, October 1996.

[13] CompuServe, Inc, "Graphics Interchange Format (GIF) Specification,"

CompuServe, Columbus, OH, 1987.

[[14]](https://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/MITRA/METHODS.HTM#REF13) Mitra, S., Wilson, M. and Sastry, K., "Performance of Multi-resolution Pattern Classifiers in Medical Image Encoding from Wavelet Coefficient Distributions," SPIE Proc. On Medical Imaging, vol 3338, February 1998.

[15] Acheroy, M. and Mangen, M. J. (1995) Progressive wavelet algorithm, versus JPEG for the compression of METOSAT data. SPIE ,Vol.2564 , pp. 64-75.

 [16] AL-Regib, Ghassan (2000) “Embedded zerotree wavelet encoding based on Shapiro paper”, ECE/Georgia Tech, Atlanta ,GA .

[17] AL-Sulaivany , Adnan. Ismail. (1998) “Digital image compression using wavelet transform”, M Sc. Thesis, college of engineering, University of Mosul,84P.

[18] Areepongsa, S.; N. Kaewkamnerd , F. Y .Syed, and K.R. Rao, (2004) “Wavelet Based Compression for Image Retrieval System”, USA:TX, University of Texas At Arlington , Box 19016,

[19] Shapiro, M. J.(1993) “Embedded image coding using zerotree of wavelet coefficients”, IEEE ,Trans. Signal Processing ,Vol.41 , No.12 , pp. 3445-3562 .

 [20] Xiong, Z.; Ramchandran, K. Ramchandran, and T. Michael. (1997) “Space frequency quantization for wavelet image coding”, IEEE, Trans, Image Processing, Vol.6, No.5, pp. 677-693.

[21] Shapiro, J. (1993). Embedded Image Coding Using Zero trees of Wavelet Coefficients, *IEEE Trans. on Signal Processing*,

41, 34453462.

[22] Dr. Vineet Richariya., Mrs. Shweta Shrivastava., NamanAgrawal. (2014). Image Compression Technique Using Different

Wavelet Function, *International Journal of Research in Computer and Communication Technology*, 3 (5) May, (Online)

2278-5841.

[23] Charles, K., Chui., Jian-Zhong Wang. (1992). On compactly supported spline wavelets and a duality principle. *Transactions*

*of the American Mathematical Society*, 330 (2) 903–915.

[24] lagendran, B., Manimurugan, S. (2012). A Survey on Various Medical Image Compression Techniques, *International*

*Journal of Soft Computing and Engineering* (IJSCE), 2 (1) March.

[25] Shravani, M., Krishna Reddy, B. (2015). IMAGE COMPRESSION USING EZW AND WDR TECHNIQUES WITH

DIFFERENT WAVELET CODES, *International Journal of Mathematical Archive*- 6 (2) Feburuary 2015.

[26] Raja, S. P., Suruliandi, A. (2010). *Performance evaluation on EZW& WDR image compression techniques*, ICCCCT, 661-664S. Bhavani, K. Thanushkodi, “A Survey on Coding Algorithms in Medical Image Compression”, International Journal on Computer Science and Engineering, Vol. 02, No. 05, pp. 1429-1434, 2010. [27] G. K. Kharate, V. H. Pati, “Color Image Compression Based On Wavelet Packet Best Tree”, International Journal of Computer Science, Vol. 7, No.3, March 2010. [28] M. R. Haque and F. Ahmed, “Image data compression with JPEG and JPEG2000”, 8th International Confrrence on Computer and Information Technology, pp. 1064-1069, 2005.

[29]A. Alice Blessie1, J. Nalini and S.C.Ramesh, “Image Compression Using Wavelet Transform Based on the Lifting Scheme and its Implementation”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011.

[30] M. J. Weinberger, G. Seroussi and G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS”, IEEE Trans. on Image Processing, Vol. 2, pp. 1309- 1324, Aug. 2000.

[31] C. M. Rahman and A. Y. Saber, “Image compression using dynamic clusturing and neural network”, 5th International Confrrence on Computer and Information Technology, pp. 453-458, December 2002.